

A map of contour integral-based eigensolvers for solving generalized eigenvalue problems

Akira Imakura^{1,*}, Lei Du², and Tetsuya Sakurai^{1,3}

¹University of Tsukuba, Japan

²Dalian University of Technology, China.

³JST/CREST

*Email : imakura@cs.tsukuba.ac.jp

Abstract

Recently, contour integral-based eigensolvers have been actively studied for solving interior eigenvalue problems that find all eigenvalues located in a certain region and their corresponding eigenvectors. In this paper, we reconsider the algorithms of the five typical contour integral-based eigensolvers from the view point of projection methods, and then map the relationships among these methods. From the analysis, we conclude that all contour integral-based eigensolvers can be regarded as projection methods and can be categorized on their subspace, an orthogonal condition and a problem to be applied implicitly.

1 Introduction

In this paper, we consider computing all eigenvalues located in a certain region of a generalized eigenvalue problem and their corresponding eigenvectors:

$$A\mathbf{x}_i = \lambda_i B\mathbf{x}_i, \quad \mathbf{x}_i \in \mathbb{C}^n \setminus \{\mathbf{0}\}, \quad \lambda_i \in \Omega \subset \mathbb{C}, \quad (1)$$

where $A, B \in \mathbb{C}^{n \times n}$ and $zB - A$ are assumed as nonsingular for any z on the boundary Γ of the region Ω . Let m be the number of target eigenvalues $\lambda_i \in \Omega$ (counting multiplicity) and $X_\Omega = [\mathbf{x}_i | \lambda_i \in \Omega]$ be a matrix whose columns are the target eigenvectors.

In 2003, Sakurai and Sugiura proposed a powerful algorithm for solving the interior eigenvalue problem (1) [15]. Their projection-type method uses certain complex moment matrices constructed by the contour integral. The basic concept is to introduce the rational function

$$r(z) := \tilde{\mathbf{v}}^H (zB - A)^{-1} B\mathbf{v}, \quad \mathbf{v}, \tilde{\mathbf{v}} \in \mathbb{C}^n \setminus \{\mathbf{0}\}, \quad (2)$$

whose poles are the eigenvalues of the generalized eigenvalue problem: $A\mathbf{x}_i = \lambda_i B\mathbf{x}_i$, and then compute all poles located in Ω by Kravanja's algorithm [12], which is based on Cauchy's integral formula.

Kravanja's algorithm can be expressed as follows. Let Γ be a positively oriented Jordan curve, i.e., the boundary of Ω . We define complex moments μ_k as

$$\mu_k := \frac{1}{2\pi i} \oint_{\Gamma} z^k r(z) dz, \quad k = 0, 1, \dots, 2M - 1.$$

Then, all poles located on Ω of an analytic function $r(z)$ are the eigenvalues of the generalized eigenvalue problem

$$H_M^< \mathbf{t}_i = \theta_i H_M \mathbf{t}_i,$$

where $H_M, H_M^<$ are Hankel matrices:

$$H_M := \begin{pmatrix} \mu_0 & \mu_1 & \cdots & \mu_{M-1} \\ \mu_1 & \mu_2 & \cdots & \mu_M \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{M-1} & \mu_M & \cdots & \mu_{2M-2} \end{pmatrix}, \quad H_M^< := \begin{pmatrix} \mu_1 & \mu_2 & \cdots & \mu_M \\ \mu_2 & \mu_3 & \cdots & \mu_{M+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_M & \mu_{M+1} & \cdots & \mu_{2M-1} \end{pmatrix}.$$

Applying Kravanja's algorithm to the rational function (2), the generalized eigenvalue problem (1) reduces to the generalized eigenvalue problem with the Hankel matrices. This algorithm is called the SS–Hankel method.

The SS–Hankel method has since been developed by several researchers. The SS–RR method based on the Rayleigh–Ritz procedure increases the accuracy of the eigenpairs [16]. Block variants of the SS–Hankel and SS–RR methods (known as the block SS–Hankel and the block SS–RR methods, respectively) improve stability of the algorithms [6–8]. The block SS–Arnoldi method based on the block Arnoldi method has also been proposed [9]. Different from these methods, Polizzi proposed the FEAST eigensolver for Hermitian generalized eigenvalue problems, which is based on an accelerated subspace iteration with the Rayleigh–Ritz procedure [13]. Their original 2009 version has been further developed [5, 19, 20].

Meanwhile, the contour integral-based eigensolvers have been extended to nonlinear eigenvalue problems. Nonlinear eigensolvers are based on the block SS–Hankel [1, 2] and the block SS–RR [21] methods and a different type of contour integral-based nonlinear eigensolver proposed by Beyn [3], which we call the Beyn method.

In this paper, we reconsider the algorithms of typical contour integral-based eigensolvers of (1), namely, the block SS–Hankel method [6, 7], the block SS–RR method [8], the FEAST eigensolver [13], the block SS–Arnoldi method [9] and the Beyn method [3] as projection methods. We then analyze and map the relationships among these methods. From the map, we also provide error analyses of each method.

The remainder of this paper is organized as follows. Sections 2 and 3 briefly describe the algorithms of the contour integral-based eigensolvers and analyze the properties of their typical matrices, respectively. The relationships among these methods are analyzed and mapped in Section 4. Error analyses of the methods are presented in Section 5, and numerical experiments are conducted in Section 6. The paper concludes with Section 7.

Throughout, the following notations are used. Let $V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L] \in \mathbb{C}^{n \times L}$ and define the range space of the matrix V by $\mathcal{R}(V) := \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L\}$. In addition, for $A \in \mathbb{C}^{n \times n}$, $\mathcal{K}_k^\square(A, V)$ and $\mathcal{B}_k^\square(A, V)$ are the block Krylov subspaces:

$$\mathcal{K}_k^\square(A, V) := \mathcal{R}([V, AV, A^2V, \dots, A^{k-1}V]),$$

$$\mathcal{B}_k^\square(A, V) := \left\{ \sum_{i=0}^{k-1} A^i V \alpha_i \mid \alpha_i \in \mathbb{C}^{L \times L} \right\}.$$

We also define a block diagonal matrix with block elements $D_i \in \mathbb{C}^{n_i \times n_i}$ constructed as

follows:

$$\bigoplus_{i=1}^d D_i = D_1 \oplus D_2 \oplus \cdots \oplus D_d = \begin{pmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_d \end{pmatrix} \in \mathbb{C}^{n \times n},$$

where $n = \sum_{i=1}^d n_i$.

2 Contour integral-based eigensolvers

The contour integral-based eigensolvers reduce the target eigenvalue problem (1) to a different type of small eigenvalue problem. In this section, we first describe the reduced eigenvalue problems and then introduce the algorithms of the contour integral-based eigensolvers.

2.1 Theoretical preparation

As a generalization of the Jordan canonical form to the matrix pencil, we have the following theorem.

Theorem 1 (Weierstrass canonical form). *Let $zB - A$ be regular. Then, there exist nonsingular matrices \tilde{P}^H, Q such that*

$$\tilde{P}^H(zB - A)Q = \bigoplus_{i=1}^r (zI_{n_i} - J_{n_i}(\lambda_i)) \oplus \bigoplus_{i=r+1}^d (zJ_{n_i}(0) - I_{n_i}),$$

where $J_{n_i}(\lambda_i)$ is the Jordan block with λ_i ,

$$J_{n_i}(\lambda) = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix} \in \mathbb{C}^{n_i \times n_i},$$

and $zJ_{n_i}(0) - I_{n_i}$ is the Jordan block with $\lambda = \infty$,

$$zJ_{n_i}(0) - I_{n_i} = \begin{pmatrix} -1 & z & & \\ & -1 & \ddots & \\ & & \ddots & z \\ & & & -1 \end{pmatrix} \in \mathbb{C}^{n_i \times n_i}.$$

The generalized eigenvalue problem $A\mathbf{x}_i = \lambda_i B\mathbf{x}_i$ has r finite eigenvalues $\lambda_i, i = 1, 2, \dots, r$ with multiplicity n_i and $d - r$ infinite eigenvalues $\lambda_i, i = r + 1, r + 2, \dots, d$ with multiplicity n_i . Let \tilde{P}_i and Q_i be submatrices of \tilde{P} and Q , respectively, corresponding to the i -th Jordan block, i.e., $\tilde{P} = [\tilde{P}_1, \tilde{P}_2, \dots, \tilde{P}_d], Q = [Q_1, Q_2, \dots, Q_d]$. Then, the columns of \tilde{P}_i and Q_i are the left/right generalized eigenvectors, whose 1st columns are the corresponding left/right eigenvectors.

Let $L, M \in \mathbb{N}$ be input parameters and $V \in \mathbb{C}^{n \times L}$ be an input matrix. We also define $S \in \mathbb{C}^{n \times LM}$ and $S_k \in \mathbb{C}^{n \times L}$ as follows:

$$S := [S_0, S_1, \dots, S_{M-1}], \quad S_k := \frac{1}{2\pi i} \oint_{\Gamma} z^k (zB - A)^{-1} BV dz. \quad (3)$$

From Theorem 1, we have the following theorem [6, 7, Theorem 4].

Theorem 2. Let $\tilde{Q}^H = Q^{-1}$ and \tilde{Q}_i be a submatrix of \tilde{Q} corresponding to the i -th Jordan block, i.e., $\tilde{Q} = [\tilde{Q}_1, \tilde{Q}_2, \dots, \tilde{Q}_d]$. Then, we have

$$S_k = Q_{\Omega} J_{\Omega}^k \tilde{Q}_{\Omega}^H V = (Q_{\Omega} J_{\Omega} \tilde{Q}_{\Omega}^H)^k (Q_{\Omega} \tilde{Q}_{\Omega}^H V) = C_{\Omega}^k S_0, \quad C_{\Omega} = Q_{\Omega} J_{\Omega} \tilde{Q}_{\Omega}^H,$$

where

$$Q_{\Omega} = [Q_i | \lambda_i \in \Omega], \quad \tilde{Q}_{\Omega} = [\tilde{Q}_i | \lambda_i \in \Omega], \quad J_{\Omega} = \bigoplus_{\lambda_i \in \Omega} J_{n_i}(\lambda_i).$$

Using Theorem 2, we also have the following theorem.

Theorem 3. Let m be the number of target eigenvalues (counting multiplicity) and $X_{\Omega} := [\mathbf{x}_i | \lambda_i \in \Omega]$ be a matrix whose columns are the target eigenvectors. Then, we have

$$\mathcal{R}(X_{\Omega}) \subset \mathcal{R}(Q_{\Omega}) = \mathcal{R}(S),$$

if and only if $\text{rank}(S) = m$.

Proof. From Theorem 2 and the definition of S , we have

$$S = [S_0, S_1, \dots, S_{M-1}] = Q_{\Omega} Z$$

where

$$Z := [(Q_{\Omega}^H V), J_{\Omega}(Q_{\Omega}^H V), \dots, J_{\Omega}^{M-1}(Q_{\Omega}^H V)].$$

Since Q_{Ω} is full rank, $\text{rank}(S) = \text{rank}(Z)$ and $\mathcal{R}(Q_{\Omega}) = \mathcal{R}(S)$ is satisfied if and only if $\text{rank}(S) = \text{rank}(Z) = m$. From the definitions of X_{Ω} and Q_{Ω} , we have $\mathcal{R}(X_{\Omega}) \subset \mathcal{R}(Q_{\Omega})$. Therefore, Theorem 3 is proven. Here, we note that $\text{rank}(Z) = m$ is not always satisfied for $m \geq LM$ even if $Q_{\Omega}^H V$ is full rank [4]. \square

2.2 Introduction to contour integral-based eigensolvers

The contour integral-based eigensolvers are mathematically designed based on Theorems 2 and 3, then the algorithms are derived from approximating the contour integral (3) using some numerical integration rule:

$$\hat{S} := [\hat{S}_0, \hat{S}_1, \dots, \hat{S}_{M-1}], \quad \hat{S}_k := \sum_{j=1}^N \omega_j z_j^k (z_j B - A)^{-1} BV, \quad (4)$$

where z_j is a quadrature point and ω_j is its corresponding weight.

2.2.1 The block SS–Hankel method

The block SS–Hankel method [6, 7] is a block variant of the SS–Hankel method. Define the block complex moments $\mu_k^\square \in \mathbb{C}^{L \times L}$ by

$$\mu_k^\square := \frac{1}{2\pi i} \oint_{\Gamma} z^k \tilde{V}^H (zB - A)^{-1} B V dz = \tilde{V}^H S_k,$$

where $\tilde{V} \in \mathbb{C}^{n \times L}$, and the block Hankel matrices $H_M, H_M^< \in \mathbb{C}^{LM \times LM}$ are given by

$$H_M^\square := \begin{pmatrix} \mu_0^\square & \mu_1^\square & \cdots & \mu_{M-1}^\square \\ \mu_1^\square & \mu_2^\square & \cdots & \mu_M^\square \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{M-1}^\square & \mu_M^\square & \cdots & \mu_{2M-2}^\square \end{pmatrix}, \quad H_M^< := \begin{pmatrix} \mu_1^\square & \mu_2^\square & \cdots & \mu_M^\square \\ \mu_2^\square & \mu_3^\square & \cdots & \mu_{M+1}^\square \\ \vdots & \vdots & \ddots & \vdots \\ \mu_M^\square & \mu_{M+1}^\square & \cdots & \mu_{2M-1}^\square \end{pmatrix}.$$

We then obtain the following theorem [6, 7, Theorem 7].

Theorem 4. *If $\text{rank}(S) = m$, then the nonsingular part of the matrix pencil $zH_M^\square - H_M^{\square<}$ is equivalent to $zI - J_\Omega$.*

According to Theorem 4, the target eigenpairs $(\lambda_i, \mathbf{x}_i), \lambda_i \in \Omega$ can be obtained through the generalized eigenvalue problem

$$H_M^{\square<} \mathbf{t}_i = \theta_i H_M^\square \mathbf{t}_i. \quad (5)$$

In practice, we approximate the block complex moments $\mu_k^\square \in \mathbb{C}^{L \times L}$ by the numerical integral (4) such that

$$\hat{\mu}_k^\square := \sum_{j=1}^N \omega_j z_j^k \tilde{V}^H (z_j B - A)^{-1} B V = \tilde{V}^H \hat{S}_k,$$

and set the block Hankel matrices $\hat{H}_M^\square, \hat{H}_M^{\square<} \in \mathbb{C}^{LM \times LM}$ as follows:

$$\hat{H}_M^\square := \begin{pmatrix} \hat{\mu}_0^\square & \hat{\mu}_1^\square & \cdots & \hat{\mu}_{M-1}^\square \\ \hat{\mu}_1^\square & \hat{\mu}_2^\square & \cdots & \hat{\mu}_M^\square \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mu}_{M-1}^\square & \hat{\mu}_M^\square & \cdots & \hat{\mu}_{2M-2}^\square \end{pmatrix}, \quad \hat{H}_M^{\square<} := \begin{pmatrix} \hat{\mu}_1^\square & \hat{\mu}_2^\square & \cdots & \hat{\mu}_M^\square \\ \hat{\mu}_2^\square & \hat{\mu}_3^\square & \cdots & \hat{\mu}_{M+1}^\square \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mu}_M^\square & \hat{\mu}_{M+1}^\square & \cdots & \hat{\mu}_{2M-1}^\square \end{pmatrix}. \quad (6)$$

To reduce the computational costs and improve the numerical stability, we also introduce a low-rank approximation with a numerical rank \hat{m} of \hat{H}_M^\square based on singular value decomposition:

$$\hat{H}_M^\square = [U_{H1}, U_{H2}] \begin{bmatrix} \Sigma_{H1} & O \\ O & \Sigma_{H2} \end{bmatrix} \begin{bmatrix} W_{H1}^H \\ W_{H2}^H \end{bmatrix} \approx U_{H1} \Sigma_{H1} W_{H1}^H.$$

In this way, the target eigenvalue problem (1) reduces to an \hat{m} dimensional standard eigenvalue problem, i.e.,

$$U_{H1}^H \hat{H}_M^{\square<} W_{H1} \Sigma_{H1}^{-1} \mathbf{t}_i = \theta_i \mathbf{t}_i.$$

The approximate eigenpairs are obtained as $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i) = (\theta_i, \hat{S} U_{H1} \mathbf{t}_i)$. The algorithm of the block SS–Hankel method is shown in Algorithm 1.

Algorithm 1 The block SS–Hankel method

Input: $L, M, N \in \mathbb{N}, V, \tilde{V} \in \mathbb{C}^{n \times L}, (z_j, \omega_j)$ for $j = 1, 2, \dots, N$

Output: Approximate eigenpairs $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i)$ for $i = 1, 2, \dots, \hat{m}$

- 1: Compute $\hat{S}_k = \sum_{j=1}^N \omega_j z_j^k (z_j B - A)^{-1} B V$ and $\hat{\mu}_k^\square = \tilde{V}^H \hat{S}_k$
 - 2: Set $\hat{S} = [\hat{S}_0, \hat{S}_1, \dots, \hat{S}_{M-1}]$ and block Hankel matrices $\hat{H}_M^\square, \hat{H}_M^{\square <}$ by (6)
 - 3: Compute SVD of \hat{H}_M^\square : $\hat{H}_M^\square = [U_{H1}, U_{H2}][\Sigma_{H1}, O; O, \Sigma_{H2}][W_{H1}, W_{H2}]^H$
 - 4: Compute eigenpairs (θ_i, \mathbf{t}_i) of $U_{H1}^H \hat{H}_M^{\square <} W_{H1}^H \Sigma_{H1}^{-1} \mathbf{t}_i = \theta_i \mathbf{t}_i$,
and compute $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i) = (\theta_i, \hat{S} U_{H1} \mathbf{t}_i)$ for $i = 1, 2, \dots, \hat{m}$
-

Algorithm 2 The block SS–RR method

Input: $L, M, N \in \mathbb{N}, V \in \mathbb{C}^{n \times L}, (z_j, \omega_j)$ for $j = 1, 2, \dots, N$

Output: Approximate eigenpairs $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i)$ for $i = 1, 2, \dots, \hat{m}$

- 1: Compute $\hat{S}_k = \sum_{j=1}^N \omega_j z_j^k (z_j B - A)^{-1} B V$, and set $\hat{S} = [\hat{S}_0, \hat{S}_1, \dots, \hat{S}_{M-1}]$
 - 2: Compute SVD of \hat{S} : $\hat{S} = [U_1, U_2][\Sigma_1, O; O, \Sigma_2][W_1, W_2]^H$
 - 3: Compute eigenpairs (θ_i, \mathbf{t}_i) of $U_1^H A U_1 \mathbf{t}_i = \theta_i U_1^H B U_1 \mathbf{t}_i$,
and compute $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i) = (\theta_i, U_1 \mathbf{t}_i)$ for $i = 1, 2, \dots, \hat{m}$
-

2.2.2 The block SS–RR method

Theorem 3 indicates that the target eigenpairs can be computed by the Rayleigh–Ritz procedure over the subspace $\mathcal{R}(S)$, i.e.,

$$S^H A S \mathbf{t}_i = \theta_i S^H B S \mathbf{t}_i.$$

The above forms the basis of the block SS–RR method [8]. In practice, the Rayleigh–Ritz procedure uses the approximated subspace $\mathcal{R}(\hat{S}) \approx \mathcal{R}(S)$ and a low-rank approximation of \hat{S} :

$$\hat{S} = [U_1, U_2] \begin{bmatrix} \Sigma_1 & O \\ O & \Sigma_2 \end{bmatrix} \begin{bmatrix} W_1^H \\ W_2^H \end{bmatrix} \approx U_1 \Sigma_1 W_1^H.$$

In this case, the reduced problem is given by

$$U_1^H A U_1 \mathbf{t}_i = \theta_i U_1^H B U_1 \mathbf{t}_i.$$

The approximate eigenpairs are obtained as $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i) = (\theta_i, U_1 \mathbf{t}_i)$. The algorithm of the block SS–RR method is shown in Algorithm 2.

2.2.3 The FEAST eigensolver

The algorithm of the accelerated subspace iteration with the Rayleigh–Ritz procedure for solving Hermitian generalized eigenvalue problems is given in Algorithm 3. Here, $\rho(A, B)$ is called an accelerator. When $\rho(A, B) = B^{-1}A$, Algorithm 3 becomes the standard subspace iteration with the Rayleigh–Ritz procedure. It computes the L largest-magnitude eigenvalues and their corresponding eigenvectors.

The FEAST eigensolver [13], proposed for Hermitian generalized eigenvalue problems, is based on an accelerated subspace iteration with the Rayleigh–Ritz procedure. In the FEAST

Algorithm 3 The accelerated subspace iteration with the Rayleigh–Ritz procedure

Input: $L \in \mathbb{N}, V_0 \in \mathbb{C}^{n \times L}$

Output: Approximate eigenpairs $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i)$ for $i = 1, 2, \dots, L$

- 1: **for** $k = 1, 2, \dots$, until convergence **do**:
 - 2: Approximate subspace projection: $Q_k = \rho(A, B) \cdot V_{k-1}$
 - 3: Compute eigenpairs $(\theta_i^{(k)}, \mathbf{t}_i^{(k)})$ of $Q_k^H A Q_k \mathbf{t}_i = \theta_i Q_k^H B Q_k \mathbf{t}_i$,
and compute $(\tilde{\lambda}_i^{(k)}, \tilde{\mathbf{x}}_i^{(k)}) = (\theta_i^{(k)}, Q_k \mathbf{t}_i^{(k)})$ for $i = 1, 2, \dots, L$
 - 4: Set $V_k = [\tilde{\mathbf{x}}_1^{(k)}, \tilde{\mathbf{x}}_2^{(k)}, \dots, \tilde{\mathbf{x}}_L^{(k)}]$
 - 5: **end for**
-

Algorithm 4 The FEAST eigensolver

Input: $L, N \in \mathbb{N}, V_0 \in \mathbb{C}^{n \times L}, (z_j, \omega_j)$ for $j = 1, 2, \dots, N$

Output: Approximate eigenpairs $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i)$ for $i = 1, 2, \dots, L$

- 1: **for** $k = 1, 2, \dots$, until convergence **do**:
 - 2: Compute $\hat{S}_0^{(k)} = \sum_{j=1}^N \omega_j (z_j B - A)^{-1} B V_{k-1}$
 - 3: Compute eigenpairs $(\theta_i^{(k)}, \mathbf{t}_i^{(k)})$ of $\hat{S}_0^{(k)H} A \hat{S}_0^{(k)} \mathbf{t}_i = \theta_i \hat{S}_0^{(k)H} B \hat{S}_0^{(k)} \mathbf{t}_i$,
and compute $(\tilde{\lambda}_i^{(k)}, \tilde{\mathbf{x}}_i^{(k)}) = (\theta_i^{(k)}, \hat{S}_0^{(k)} \mathbf{t}_i^{(k)})$ for $i = 1, 2, \dots, L$
 - 4: Set $V_k = [\tilde{\mathbf{x}}_1^{(k)}, \tilde{\mathbf{x}}_2^{(k)}, \dots, \tilde{\mathbf{x}}_L^{(k)}]$
 - 5: **end for**
-

eigensolver, the accelerator $\rho(A, B)$ is set as

$$\rho(A, B) = \sum_{j=1}^N \omega_j (z_j B - A)^{-1} B \approx \frac{1}{2\pi i} \oint_{\Gamma} (z B - A)^{-1} B dz,$$

based on Theorem 3. Therefore, the FEAST eigensolver computes the eigenvalues located on Ω and their corresponding eigenvectors. For numerical integration, the FEAST eigensolver uses the Gauß-Legendre quadrature or the Zolotarev quadrature; see [5, 13].

In each iteration of the FEAST eigensolver, the target eigenvalue problem (1) is reduced to a small eigenvalue problem, i.e.,

$$\hat{S}_0^H A \hat{S}_0 \mathbf{t}_i = \theta_i \hat{S}_0^H B \hat{S}_0 \mathbf{t}_i,$$

based on the Rayleigh–Ritz procedure. The approximate eigenpairs are obtained as $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i) = (\theta_i, \hat{S}_0 \mathbf{t}_i)$. The algorithm of the FEAST eigensolver is shown in Algorithm 4.

2.2.4 The block SS–Arnoldi method

From Theorems 2 and 3 and the definition of $C_\Omega := Q_\Omega J_\Omega \tilde{Q}_\Omega^H$, we have the following three theorems [9].

Theorem 5. *The subspace $\mathcal{R}(S)$ can be expressed as the block Krylov subspace associated with the matrix C_Ω :*

$$\mathcal{R}(S) = \mathcal{K}_M^\square(C_\Omega, S_0).$$

Theorem 6. Let m be the number of target eigenvalues (counting multiplicity). Then, if $\text{rank}(S) = m$, the target eigenvalue problem (1) is equivalent to a standard eigenvalue problem of the form

$$C_\Omega \mathbf{x}_i = \lambda_i \mathbf{x}_i, \quad \mathbf{x}_i \in \mathcal{R}(S) = \mathcal{K}_M^\square(C_\Omega, S_0). \quad (7)$$

Theorem 7. Any $E_k \in \mathcal{B}_k^\square(C_\Omega, S_0)$ has the following formula:

$$E_k = \frac{1}{2\pi i} \oint_\Gamma \sum_{i=0}^{k-1} z^i (zB - A)^{-1} BV \alpha_i dz, \quad \alpha_i \in \mathbb{C}^{L \times L}.$$

Then, the matrix multiplication of C_Ω by E_k becomes

$$C_\Omega E_k = \frac{1}{2\pi i} \oint_\Gamma z \sum_{i=0}^{k-1} z^i (zB - A)^{-1} BV \alpha_i dz.$$

From Theorems 5 and 6, we observe that the target eigenpairs $(\lambda_i, \mathbf{x}_i)$, $\lambda_i \in \Omega$ can be computed by the block Arnoldi method with the block Krylov subspace $\mathcal{K}_M^\square(C_\Omega, S_0)$ for solving the standard eigenvalue problem (7). Here, we note that the matrix C_Ω is not explicitly constructed. Instead, the matrix multiplication of C_Ω can be computed via the contour integral using Theorem 7. By approximating the contour integral by a numerical integration rule, the algorithm of the block SS–Arnoldi method is derived (Algorithm 5).

A low-rank approximation technique to reduce the computational costs and improve stability is not applied in the current version of the block SS–Arnoldi method [9]. Improvements of the block SS–Arnoldi method has been developed in [10].

2.2.5 The Beyn method

The Beyn method is a nonlinear eigensolver based on the contour integral [3]. In this subsection, we consider the algorithm of the Beyn method for solving the generalized eigenvalue problem (1).

Let the singular value decomposition of S_0 be $S_0 = U_0 \Sigma_0 W_0^H$. Then, from Theorem 2, the target eigenpairs $(\lambda_i, \mathbf{x}_i)$, $\lambda_i \in \Omega$ are computed by solving

$$U_0^H S_1 W_0 \Sigma_0^{-1} \mathbf{t}_i = \theta_i \mathbf{t}_i,$$

where $(\lambda_i, \mathbf{x}_i) = (\theta_i, U_0 \mathbf{t}_i)$ [3]. In practice, we compute a low-rank approximation of \widehat{S}_0 by the singular value decomposition, i.e.,

$$\widehat{S}_0 = [U_{0,1}, U_{0,2}] \begin{bmatrix} \Sigma_{0,1} & O \\ O & \Sigma_{0,2} \end{bmatrix} \begin{bmatrix} W_{0,1}^H \\ W_{0,2}^H \end{bmatrix} \approx U_{0,1} \Sigma_{0,1} W_{0,1}^H. \quad (8)$$

which reduces the target eigenvalue problem (1) to the standard eigenvalue problem

$$U_{0,1}^H \widehat{S}_1 W_{0,1} \Sigma_{0,1}^{-1} \mathbf{t}_i = \theta_i \mathbf{t}_i. \quad (9)$$

The approximate eigenpairs are obtained as $(\widetilde{\lambda}_i, \widetilde{\mathbf{x}}_i) = (\theta_i, U_{0,1} \mathbf{t}_i)$. The algorithm of the Beyn method for solving the generalized eigenvalue problem (1) is shown in Algorithm 6.

Algorithm 5 The block SS–Arnoldi method

Input: $L, M, N \in \mathbb{N}, V \in \mathbb{C}^{n \times L}, (z_j, \omega_j)$ for $j = 1, 2, \dots, N$

Output: Approximate eigenpairs $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i)$ for $i = 1, 2, \dots, LM$

- 1: Solve $Y_j = (z_j B - A)^{-1} B V$ for $j = 1, 2, \dots, N$
 - 2: $W_0 = \sum_{j=1}^N \omega_j Y_j$
 - 3: Compute QR decomposition of W_0 : $W_0 = W_1 R$
 - 4: Set $\alpha_{1,j} = R^{-1}$ for $j = 1, 2, \dots, N$
 - 5: **for** $k = 1, 2, \dots, M$ **do**:
 - 6: $\tilde{\alpha}_{k,j} = z_j \alpha_{k,j}$ for $j = 1, 2, \dots, N$
 - 7: $\tilde{W}_k = \sum_{j=1}^N \omega_j Y_j \tilde{\alpha}_{k,j}$
 - 8: **for** $i = 1, 2, \dots, k$ **do**:
 - 9: $H_{i,k} = W_i^H \tilde{W}_k$
 - 10: $\tilde{\alpha}_{k,j} = \tilde{\alpha}_{k,j} - \alpha_{i,j} H_{i,k}$ for $j = 1, 2, \dots, N$
 - 11: $\tilde{W}_k = \tilde{W}_k - W_i H_{i,k}$
 - 12: **end for**
 - 13: Compute QR decomposition of \tilde{W}_k : $\tilde{W}_k = W_{k+1} H_{k+1,k}$
 - 14: $\alpha_{k+1,j} = \tilde{\alpha}_{k,j} H_{k+1,k}^{-1}$ for $j = 1, 2, \dots, N$
 - 15: **end for**
 - 16: Set $W = [W_1, W_2, \dots, W_M]$ and $H_M = \{H_{i,j}\}_{1 \leq i,j \leq M}$
 - 17: Compute eigenpairs (θ_i, \mathbf{t}_i) of $H_M \mathbf{t}_i = \theta_i \mathbf{t}_i$,
and compute $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i) = (\theta_i, W \mathbf{t}_i)$ for $i = 1, 2, \dots, LM$
-

Algorithm 6 The Beyn method

Input: $L, N \in \mathbb{N}, V \in \mathbb{C}^{n \times L}, (z_j, \omega_j)$ for $j = 1, 2, \dots, N$

Output: Approximate eigenpairs $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i)$ for $i = 1, 2, \dots, \hat{m}$

- 1: Compute \hat{S}_0, \hat{S}_1 , where $\hat{S}_k = \sum_{j=1}^N \omega_j z_j^k (z_j B - A)^{-1} B V$
 - 2: Compute SVD of \hat{S}_0 : $\hat{S}_0 = [U_{0,1}, U_{0,2}] [\Sigma_{0,1}, O; O, \Sigma_{0,2}] [W_{0,1}, W_{0,2}]^H$
 - 3: Compute eigenpairs (θ_i, \mathbf{t}_i) of $U_{0,1}^H \hat{S}_1 W_{0,1} \Sigma_{0,1}^{-1} \mathbf{t}_i = \theta_i \mathbf{t}_i$,
and compute $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i) = (\theta_i, U_{0,1} \mathbf{t}_i)$ for $i = 1, 2, \dots, \hat{m}$
-

3 Theoretical preliminaries for map building

As shown in Section 2, contour integral-based eigensolvers are based on the property of the matrices S and S_k (Theorems 2 and 3). The practical algorithms are then derived by a numerical integral approximation. As theoretical preliminaries for map building in Section 4, this section explores the properties of the approximated matrices \hat{S} and \hat{S}_k . Here, we assume that (z_j, ω_j) satisfy the following condition:

$$\sum_{j=1}^N \omega_j z_j^k \begin{cases} = 0, & k = 0, 1, \dots, N-2 \\ \neq 0, & k = -1 \end{cases}. \quad (10)$$

If the matrix pencil $zB - A$ is diagonalizable and (z_j, ω_j) satisfies condition (10), we have

$$\hat{S}_k = C^k \hat{S}_0, \quad C = X_r \Lambda_r \tilde{X}_r^H,$$

where $X_r = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r]$ is a matrix whose columns are eigenvectors corresponding to finite eigenvalues, $\tilde{X}_r = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_r]$ is a submatrix of $\tilde{X} = X^{-H}$: $\tilde{X}_r^H X_r = I$, and $\Lambda_r = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_r)$; see [11]. In the following analysis, we introduce a similar relationship in the case that the matrix pencil $zB - A$ is non-diagonalizable. First, we define an upper triangular Toeplitz matrix as follows.

Definition 1. For $\mathbf{a} = [a_1, a_2, \dots, a_n] \in \mathbb{C}^{1 \times n}$, define $T_n(\mathbf{a})$ as an $n \times n$ triangular Toeplitz matrix, i.e.,

$$T_n(\mathbf{a}) := \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ 0 & a_1 & \cdots & a_{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_1 \end{pmatrix} \in \mathbb{C}^{n \times n}.$$

Let $\mathbf{a} = [a_1, a_2, \dots, a_n]$, $\mathbf{b} = [b_1, b_2, \dots, b_n]$, $\mathbf{c} = [c_1, c_2, \dots, c_n] \in \mathbb{C}^{1 \times n}$ and $\alpha, \beta \in \mathbb{C}$. Then, we have

$$\alpha T_n(\mathbf{a}) + \beta T_n(\mathbf{b}) = T_n(\alpha \mathbf{a} + \beta \mathbf{b}), \quad (11)$$

$$T_n(\mathbf{a})T_n(\mathbf{b}) = T_n(\mathbf{c}), \quad c_i = \sum_{j=1}^i a_j b_{i-j+1}, \quad i = 1, 2, \dots, n. \quad (12)$$

Letting $\mathbf{d} = [\alpha, \beta, 0, \dots, 0] \in \mathbb{C}^{1 \times n}$, we also have

$$(T_n(\mathbf{d}))^k = T_n(\mathbf{d}^{(k)}), \quad \mathbf{d}^{(k)} = \left[\binom{k}{0} \alpha^k \beta^0, \binom{k}{1} \alpha^{k-1} \beta^1, \dots, \binom{k}{n} \alpha^{k-n+1} \beta^{n-1} \right], \quad (13)$$

$$(T_n(\mathbf{d}))^{-1} = T_n(\mathbf{d}^{(-1)}), \quad \mathbf{d}^{(-1)} = \left[\frac{1}{\alpha}, -\frac{\beta}{\alpha^2}, \dots, \frac{(-\beta)^{n-1}}{\alpha^n} \right]. \quad (14)$$

Using these relations (11)–(14), we analyze properties of \hat{S} and \hat{S}_k . From Theorem 1, we have

$$(zB - A)^{-1} = Q \left[\bigoplus_{i=1}^r (zI_{n_i} - J_{n_i}(\lambda_i))^{-1} \oplus \bigoplus_{i=r+1}^d (zJ_{n_i}(0) - I_{n_i})^{-1} \right] \tilde{P}^H, \\ B = P \left[\bigoplus_{i=1}^r I_{n_i} \oplus \bigoplus_{i=r+1}^d J_{n_i}(0) \right] \tilde{Q}^H,$$

where $P := \tilde{P}^{-H}$, $\tilde{Q}^H := Q^{-1}$. Therefore, the matrix \hat{S}_k can be written as

$$\begin{aligned} \hat{S}_k &= \sum_{j=1}^N \omega_j z_j^k (z_j B - A)^{-1} B V \\ &= \sum_{j=1}^N \omega_j z_j^k Q \left\{ \bigoplus_{i=1}^r (z_j I_{n_i} - J_{n_i}(\lambda_i))^{-1} \oplus \bigoplus_{i=r+1}^d [(z_j J_{n_i}(0) - I_{n_i})^{-1} J_{n_i}(0)] \right\} \tilde{Q}^H V \\ &= \left\{ \sum_{i=1}^r Q_i \left[\sum_{j=1}^N \omega_j z_j^k (z_j I_{n_i} - J_{n_i}(\lambda_i))^{-1} \right] \tilde{Q}_i^H V \right\} \\ &\quad + \left\{ \sum_{i=r+1}^d Q_i \left[\sum_{j=1}^N \omega_j z_j^k (z_j J_{n_i}(0) - I_{n_i})^{-1} J_{n_i}(0) \right] \tilde{Q}_i^H V \right\}, \end{aligned}$$

where Q_i and \tilde{Q}_i are $n \times n_i$ submatrices of Q and \tilde{Q} respectively, corresponding to the i -th Jordan block, i.e., $Q = [Q_1, Q_2, \dots, Q_d]$, $\tilde{Q} = [\tilde{Q}_1, \tilde{Q}_2, \dots, \tilde{Q}_d]$.

First, we consider the 1st term of \hat{S}_k :

$$\hat{S}_k^{(1)} := \sum_{i=1}^r Q_i \left[\sum_{j=1}^N \omega_j z_j^k (z_j I_{n_i} - J_{n_i}(\lambda_i))^{-1} \right] \tilde{Q}_i^H V.$$

From the relation

$$z_j I_{n_i} - J_{n_i}(\lambda_i) = T_{n_i}([z_j - \lambda_i, -1, 0, \dots, 0])$$

and (14), we have

$$(z_j I - J_{n_i}(\lambda_i))^{-1} = T_{n_i} \left(\left[\frac{1}{z_j - \lambda_i}, \frac{1}{(z_j - \lambda_i)^2}, \dots, \frac{1}{(z_j - \lambda_i)^{n_i}} \right] \right).$$

Thus, defining $\mathbf{f}_k(\lambda_i) \in \mathbb{C}^{1 \times n_i}$ as

$$\mathbf{f}_k(\lambda_i) := \sum_{j=1}^N \omega_j z_j^k \left[\frac{1}{z_j - \lambda_i}, \frac{1}{(z_j - \lambda_i)^2}, \dots, \frac{1}{(z_j - \lambda_i)^{n_i}} \right], \quad (15)$$

from (11), we have

$$\sum_{j=1}^N \omega_j z_j^k (z_j I_{n_i} - J_{n_i}(\lambda_i))^{-1} = T_{n_i}(\mathbf{f}_k(\lambda_i)).$$

Here, the following propositions hold.

Proposition 1. Suppose that (ω_j, z_j) satisfies condition (10). Then, for any $\lambda_i \neq 0$ and $0 \leq k \leq N - p, p \geq 1$, the relation

$$\sum_{j=1}^N \frac{\omega_j z_j^k}{(z_j - \lambda_i)^p} = \lambda_i^k \sum_{j=1}^N \frac{\omega_j}{(z_j - \lambda_i)^p} \sum_{q=0}^{p-1} \binom{k}{q} \left(\frac{z_j - \lambda_i}{\lambda_i} \right)^q \quad (16)$$

is satisfied.

Proof. Since $\lambda_i \neq 0$, we have

$$\frac{\omega_j z_j^k}{(z_j - \lambda_i)^p} = \frac{\omega_j}{(z_j - \lambda_i)^p} \lambda_i^k \left(\frac{z_j}{\lambda_i} \right)^k = \frac{\omega_j}{(z_j - \lambda_i)^p} \lambda_i^k \left(1 + \frac{z_j - \lambda_i}{\lambda_i} \right)^k. \quad (17)$$

Here, from the binomial theorem $(a + b)^k = \sum_{q=0}^k \binom{k}{q} a^{k-q} b^q$, (17) is rewritten by

$$\frac{\omega_j z_j^k}{(z_j - \lambda_i)^p} = \frac{\omega_j}{(z_j - \lambda_i)^p} \lambda_i^k \sum_{q=0}^k \binom{k}{q} \left(\frac{z_j - \lambda_i}{\lambda_i} \right)^q.$$

Therefore, the left term of (16) is

$$\begin{aligned} \sum_{j=1}^N \frac{\omega_j z_j^k}{(z_j - \lambda_i)^p} &= \sum_{j=1}^N \frac{\omega_j}{(z_j - \lambda_i)^p} \lambda_i^k \sum_{q=0}^k \binom{k}{q} \left(\frac{z_j - \lambda_i}{\lambda_i} \right)^q \\ &= \lambda_i^k \sum_{q=0}^k \binom{k}{q} \lambda_i^{-q} \sum_{j=1}^N \omega_j (z_j - \lambda_i)^{q-p}. \end{aligned}$$

Because condition (10) is satisfied, we have

$$\sum_{j=1}^N \omega_j (z_j - \lambda_i)^{q-p} = 0, \quad q = 1, 2, \dots, N-p.$$

Therefore, for $k = 0, 1, \dots, N-p$, we obtain

$$\sum_{j=1}^N \frac{\omega_j z_j^k}{(z_j - \lambda_i)^p} = \lambda_i^k \binom{k}{0} \lambda_i^{-0} \sum_{j=1}^N \omega_j (z_j - \lambda_i)^{-p} = \lambda_i^k \sum_{j=1}^N \frac{\omega_j}{(z_j - \lambda_i)^p},$$

which proves Proposition 1. \square

Proposition 2. Suppose that (ω_j, z_j) satisfies condition (10). Then, for any $0 \leq k \leq N - n_i$, the relation

$$T_{n_i}(\mathbf{f}_k(\lambda_i)) = (J_{n_i}(\lambda_i))^k T_{n_i}(\mathbf{f}_0(\lambda_i)) \quad (18)$$

is satisfied, where $\mathbf{f}_k(\lambda_i)$ is defined by (15) and $0^0 = 1$.

Proof. We first consider the case of $\lambda_i = 0$. From the relation $J_{n_i}(0) = T_{n_i}([0, 1, 0, \dots, 0])$, there exists a vector $\mathbf{t}_{0,k} \in \mathbb{C}^{1 \times n_i}$ satisfying

$$T_{n_i}(\mathbf{t}_{0,k}) := (J_{n_i}(0))^k T_{n_i}(\mathbf{f}_0(0)).$$

Then, from (12) and (13), the p -th element $(\mathbf{t}_{0,k})_p$ of $\mathbf{t}_{0,k}$ can be written as

$$(\mathbf{t}_{0,k})_p = \begin{cases} 0 & (0 \leq p \leq k) \\ \sum_{j=1}^N \omega_j z_j^{k-p} & (k \leq p) \end{cases}.$$

Moreover, the vector $\mathbf{f}_k(0)$ can be written as

$$\mathbf{f}_k(0) = \sum_{j=1}^N \omega_j [z_j^{k-1}, z_j^{k-2}, \dots, z_j^{k-n_i}],$$

and from condition (10), we have $\mathbf{f}_k(0) = \mathbf{t}_{0,k}$. Therefore, (18) is satisfied for $\lambda_i = 0$.

Next, we consider the case of $\lambda_i \neq 0$. From $J_{n_i}(\lambda_i) = T_{n_i}([\lambda_i, 1, 0, \dots, 0])$ and (13), we have

$$(J_{n_i}(\lambda_i))^k = T_{n_i} \left(\left[\lambda_i^k, \binom{k}{1} \lambda_i^{k-1}, \dots, \binom{k}{n_i} \lambda_i^{k-n_i+1} \right] \right).$$

Let $\mathbf{t}_k \in \mathbb{C}^{1 \times n_i}$ be a vector satisfying

$$T_{n_i}(\mathbf{t}_k) := (J_{n_i}(\lambda_i))^k T_{n_i}(\mathbf{f}_0(\lambda_i)).$$

Then, the p -th element $(\mathbf{t}_k)_p$ of \mathbf{t}_k can be written as

$$\begin{aligned} (\mathbf{t}_k)_p &= \sum_{q=1}^p \binom{k}{q-1} \lambda_i^{k-q+1} \sum_{j=1}^N \omega_j \frac{1}{(z_j - \lambda_i)^{p-q+1}} \\ &= \lambda_i^k \sum_{j=1}^N \frac{\omega_j}{(z_j - \lambda_i)^p} \sum_{q=1}^p \binom{k}{q-1} \frac{(z_j - \lambda_i)^{q-1}}{\lambda_i^{q-1}} \\ &= \lambda_i^k \sum_{j=1}^N \frac{\omega_j}{(z_j - \lambda_i)^p} \sum_{q=0}^{p-1} \binom{k}{q} \left(\frac{z_j - \lambda_i}{\lambda_i} \right)^q. \end{aligned}$$

By Proposition 1, for $0 \leq k \leq N - n_i$, we obtain

$$(\mathbf{t}_k)_p = \sum_{j=1}^N \frac{\omega_j z_j^k}{(z_j - \lambda)^p}.$$

Therefore, we have

$$\mathbf{t}_k = \mathbf{f}_k(\lambda_i),$$

and

$$T_{n_i}(\mathbf{f}_k(\lambda_i)) = T_{n_i}(\mathbf{t}_k) = (J_{n_i}(\lambda_i))^k T_{n_i}(\mathbf{f}_0(\lambda_i))$$

is satisfied, proving Proposition 2. \square

From Proposition 2, let

$$\eta_1 := \max_{1 \leq i \leq r} n_i,$$

and let (z_j, ω_j) satisfy condition (10). The 1st term of \hat{S}_k then becomes

$$\hat{S}_k^{(1)} = \sum_{i=1}^r Q_i (J_{n_i}(\lambda_i))^k \left[\sum_{j=1}^N \omega_j (z_j I_{n_i} - J_{n_i}(\lambda_i))^{-1} \right] \tilde{Q}_i^H V \quad (19)$$

for any $0 \leq k \leq N - \eta_1$.

Now consider the 2nd term of \hat{S}_k , i.e.,

$$\hat{S}_k^{(2)} := \sum_{i=r+1}^d Q_i \left[\sum_{j=1}^N \omega_j z_j^k (z_j J_{n_i}(0) - I_{n_i})^{-1} J_{n_i}(0) \right] \tilde{Q}_i^H V.$$

From the relations

$$\begin{aligned} z_j J_{n_i}(0) - I_{n_i} &= T_{n_i}([-1, z_j, 0, \dots, 0]), \\ J_{n_i}(0) &= T_{n_i}([0, 1, 0, \dots, 0]) \end{aligned}$$

and (12) and (14), we have

$$(z_j J_{n_i}(0) - I)^{-1} J_{n_i}(0) = -T_{n_i}([0, 1, z_j, z_j^2, \dots, z_j^{n_i-2}]).$$

In Addition, from (11), we have

$$\begin{aligned} & \sum_{j=1}^N \omega_j z_j^k (z_j J_{n_i}(0) - I_{n_i})^{-1} J_{n_i}(0) \\ &= -T_{n_i} \left(\left[0, \sum_{j=1}^N \omega_j z_j^k, \sum_{j=1}^N \omega_j z_j^{k+1}, \dots, \sum_{j=1}^N \omega_j z_j^{k+n_i-2} \right] \right). \end{aligned}$$

Here, because (z_j, ω_j) satisfies condition (10),

$$\sum_{j=1}^N \omega_j z_j^k (z_j J_{n_i}(0) - I_{n_i})^{-1} J_{n_i}(0) = O$$

is satisfied for any $0 \leq k \leq N - n_i$. Therefore, letting

$$\eta_2 := \max_{r+1 \leq i \leq d} n_i,$$

the 2nd term of \widehat{S}_k is O for any $0 \leq k \leq N - \eta_2$, i.e.,

$$\widehat{S}_k^{(2)} = O. \quad (20)$$

From (19) and (20), we have the following theorems.

Theorem 8. *Let η be the maximum size of the Jordan blocks:*

$$\eta = \max_{1 \leq i \leq d} n_i.$$

Then, if (z_j, ω_j) satisfies condition (10), we have

$$\widehat{S}_k = C^k \widehat{S}_0, \quad C = Q_{1:r} J_{1:r} \widetilde{Q}_{1:r}^H,$$

for any $0 \leq k \leq N - \eta$, where

$$Q_{1:r} := [Q_1, Q_2, \dots, Q_r], \quad \widetilde{Q}_{1:r} := [\widetilde{Q}_1, \widetilde{Q}_2, \dots, \widetilde{Q}_r], \quad J_{1:r} := \bigoplus_{i=1}^r J_{n_i}(\lambda_i).$$

Proof. Since $\eta = \max(\eta_1, \eta_2)$, from (19) and (20), we have

$$\widehat{S}_k = \sum_{i=1}^r Q_i (J_{n_i}(\lambda_i))^k \left[\sum_{j=1}^N \omega_j (z_j I_{n_i} - J_{n_i}(\lambda_i))^{-1} \right] \widetilde{Q}_i^H V,$$

for any $0 \leq k \leq N - \eta$. Here, we let

$$F_{n_i} := T_{n_i}(\mathbf{f}_0(\lambda_i)) = \sum_{j=1}^N \omega_j (z_j I_{n_i} - J_{n_i}(\lambda_i))^{-1},$$

$$F_{1:r} := \bigoplus_{i=1}^r F_{n_i},$$

then we obtain

$$\begin{aligned} \widehat{S}_k &= \sum_{i=1}^r Q_i (J_{n_i}(\lambda_i))^k F_{n_i} \widetilde{Q}_i^H V \\ &= Q_{1:r} J_{1:r}^k F_{1:r} \widetilde{Q}_{1:r}^H V \\ &= (Q_{1:r} J_{1:r} \widetilde{Q}_{1:r}^H)^k (Q_{1:r} F_{1:r} \widetilde{Q}_{1:r}^H V) \\ &= C^k \widehat{S}_0. \end{aligned}$$

Therefore, Theorem 8 is proven. \square

Theorem 9. *If (z_j, ω_j) satisfies condition (10), then the standard eigenvalue problem*

$$C \mathbf{x}_i = \lambda_i \mathbf{x}_i, \quad \mathbf{x}_i \in \mathcal{R}(Q_{1:r}), \quad \lambda_i \in \Omega \subset \mathbb{C}, \quad (21)$$

is equivalent to the generalized eigenvalue problem (1).

Proof. From the definition of $C := Q_{1:r} J_{1:r} \widetilde{Q}_{1:r}^H$, the matrix C has the same right eigenpairs $(\lambda_i, \mathbf{x}_i)$, $i = 1, 2, \dots, r$ as the matrix pencil $zB - A$, i.e., $\mathbf{x}_i \in \mathcal{R}(Q_{1:r})$. The other eigenvalues of C are 0, and their corresponding eigenvectors are equivalent to the right eigenvectors associated with the infinite eigenvalues $\lambda_i = \infty$ of $zB - A$, i.e., $\mathbf{x}_i \notin \mathcal{R}(Q_{1:r})$. Therefore, Theorem 9 is proven. \square

4 A map of the contour integral-based eigensolvers

Section 3 analyzed the properties of the approximated matrices \hat{S} and \hat{S}_k (Theorem 8) and introduced the standard eigenvalue problem (21) equivalent to the target eigenvalue problem (1) (Theorem 9).

In this section, based on Theorems 8 and 9, we reconsider the algorithms of the contour integral-based eigensolvers in terms of projection methods and map the relationships, focusing on the subspaces, orthogonal conditions and problems to be applied.

4.1 Reconsideration of the contour integral-based eigensolvers

As described in Section 2, the subspaces $\mathcal{R}(S)$ and $\mathcal{R}(S_k)$ contain only the target eigenvectors $\mathbf{x}_i, \lambda_i \in \Omega$ based on Cauchy's integral formula. In constant, the subspaces $\mathcal{R}(\hat{S})$ and $\mathcal{R}(\hat{S}_k)$ are rich in the component of the target eigenvectors as will be shown in Section 5.

4.1.1 The block SS–RR method and the FEAST eigensolvers

The block SS–RR method and the FEAST eigensolvers are easily reconfigured as projection methods.

The block SS–RR method solves $A\mathbf{x}_i = \lambda_i B\mathbf{x}_i$ through the Rayleigh–Ritz procedure on $\mathcal{R}(\hat{S})$. The block SS–RR method (Algorithm 2) is derived using a low-rank approximation of the matrix \hat{S} as shown in Section 2.2. Since $\mathcal{R}(\hat{S})$ is rich in the component of the target eigenvectors, the target eigenpairs are well approximated by the Rayleigh–Ritz procedure.

The FEAST eigensolver conducts accelerated subspace iteration with the Rayleigh–Ritz procedure. In each iteration of the FEAST eigensolver, the Rayleigh–Ritz procedure solves $A\mathbf{x}_i = \lambda_i B\mathbf{x}_i$ on $\mathcal{R}(\hat{S}_0)$. Like $\mathcal{R}(\hat{S})$ in the block SS–RR method, $\mathcal{R}(\hat{S}_0)$ is rich in the component of the target eigenvectors; therefore, the FEAST eigensolver also well approximates the target eigenpairs by the Rayleigh–Ritz procedure.

4.1.2 The block SS–Hankel method, the block SS–Arnoldi method and the Beyn method

From Theorem 8, we rewrite the block complex moments $\hat{\mu}_k^\square$ of the block SS–Hankel method as

$$\hat{\mu}_k^\square = \tilde{V}^H \hat{S}_k = \tilde{V}^H C \hat{S}_{k-1} = \cdots = \tilde{V}^H C^k \hat{S}_0.$$

Thus, the block Hankel matrices $\hat{H}_M^\square, \hat{H}_M^{\square<}$ become

$$\hat{H}_M^\square = \begin{pmatrix} \tilde{V}^H \hat{S}_0 & \tilde{V}^H \hat{S}_1 & \cdots & \tilde{V}^H \hat{S}_{M-1} \\ \tilde{V}^H C \hat{S}_0 & \tilde{V}^H C \hat{S}_1 & \cdots & \tilde{V}^H C \hat{S}_{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{V}^H C^{M-1} \hat{S}_0 & \tilde{V}^H C^{M-1} \hat{S}_1 & \cdots & \tilde{V}^H C^{M-1} \hat{S}_{M-1} \end{pmatrix},$$

$$\hat{H}_M^{\square<} = \begin{pmatrix} \tilde{V}^H \hat{S}_0 & \tilde{V}^H \hat{S}_1 & \cdots & \tilde{V}^H \hat{S}_{M-1} \\ \tilde{V}^H C \hat{S}_0 & \tilde{V}^H C \hat{S}_1 & \cdots & \tilde{V}^H C \hat{S}_{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{V}^H C^{M-1} \hat{S}_0 & \tilde{V}^H C^{M-1} \hat{S}_1 & \cdots & \tilde{V}^H C^{M-1} \hat{S}_{M-1} \end{pmatrix},$$

respectively. Here, let

$$\tilde{S} := [\tilde{V}, C^H \tilde{V}, (C^H)^2 \tilde{V}, \dots, (C^H)^{M-1} \tilde{V}].$$

Then, we have

$$\hat{H}_M^\square = \tilde{S}^H \hat{S}, \quad \hat{H}_M^{\square<} = \tilde{S}^H C \hat{S}.$$

Therefore, the generalized eigenvalue problem (5) is rewritten as

$$\tilde{S}^H C \hat{S} \mathbf{t}_i = \theta_i \tilde{S}^H \hat{S} \mathbf{t}_i. \quad (22)$$

In this form, the block SS–Hankel method can be regarded as a Petrov–Galerkin-type projection method for solving the standard eigenvalue problem (21), i.e., the approximate solution $\tilde{\mathbf{x}}_i$ and the corresponding residual $\mathbf{r}_i := C\tilde{\mathbf{x}}_i - \theta_i \tilde{\mathbf{x}}_i$ satisfy $\tilde{\mathbf{x}}_i \in \mathcal{R}(\hat{S})$ and $\mathbf{r}_i \perp \mathcal{R}(\tilde{S})$, respectively. Recognizing that $\mathcal{R}(\hat{S}) \subset \mathcal{R}(Q_{1:r})$ and applying Theorem 9, we find that the block SS–Hankel method obtains the target eigenpairs.

Since the Petrov–Galerkin-type projection method for (21) does not perform the (bi-)orthogonalization; that is $\tilde{S}^H \hat{S} \neq I$, (22) describes the generalized eigenvalue problem. The practical algorithm of the block SS–Hankel method (Algorithm 1) is derived from a low-rank approximation of (22).

From Theorem 8, we have

$$\mathcal{R}(\hat{S}) = \mathcal{K}_M^\square(C, \hat{S}_0)$$

similar to Theorem 5. Therefore, the block SS–Arnoldi method can be regarded as a block Arnoldi method with $\mathcal{K}_M^\square(C, \hat{S}_0)$ for solving the standard eigenvalue problem (21). Moreover, for $M \leq N - \eta$, any $\hat{E}_M \in \mathcal{B}_M^\square(C, \hat{S}_0)$ can be written as

$$\hat{E}_M = \sum_{j=1}^N \omega_j \sum_{i=0}^{M-1} z_j^i (z_j B - A)^{-1} B V \alpha_i, \quad \alpha_i \in \mathbb{C}^{L \times L}.$$

and the matrix multiplication of C by \hat{E}_M is given by

$$C \hat{E}_M = \sum_{j=1}^N \omega_j z_j \sum_{i=0}^{M-1} z_j^i (z_j B - A)^{-1} B V \alpha_i.$$

similar to Theorem 7. Therefore, in each iteration, the matrix multiplication of C can be performed by a numerical integration.

The Beyn method can be also regarded as a projection method for solving the standard eigenvalue problem (21). From the relation $\hat{S}_1 = C \hat{S}_0$ and the singular value decomposition (8) of \hat{S}_0 , the coefficient matrix of the eigenvalue problem (9) obtained from the Beyn method becomes

$$U_{0,1}^H \hat{S}_1 W_{0,1} \Sigma_1^{-1} = U_{0,1}^H C \hat{S}_0 W_{0,1} \Sigma_{0,1}^{-1} = U_{0,1}^H C U_{0,1}.$$

Therefore, the Byen method can be regarded as a Rayleigh–Ritz-type projection method on $\mathcal{R}(U_{0,1})$ for solving (21), where $\mathcal{R}(U_{0,1})$ is obtained from a low-rank approximation of \hat{S}_0 .

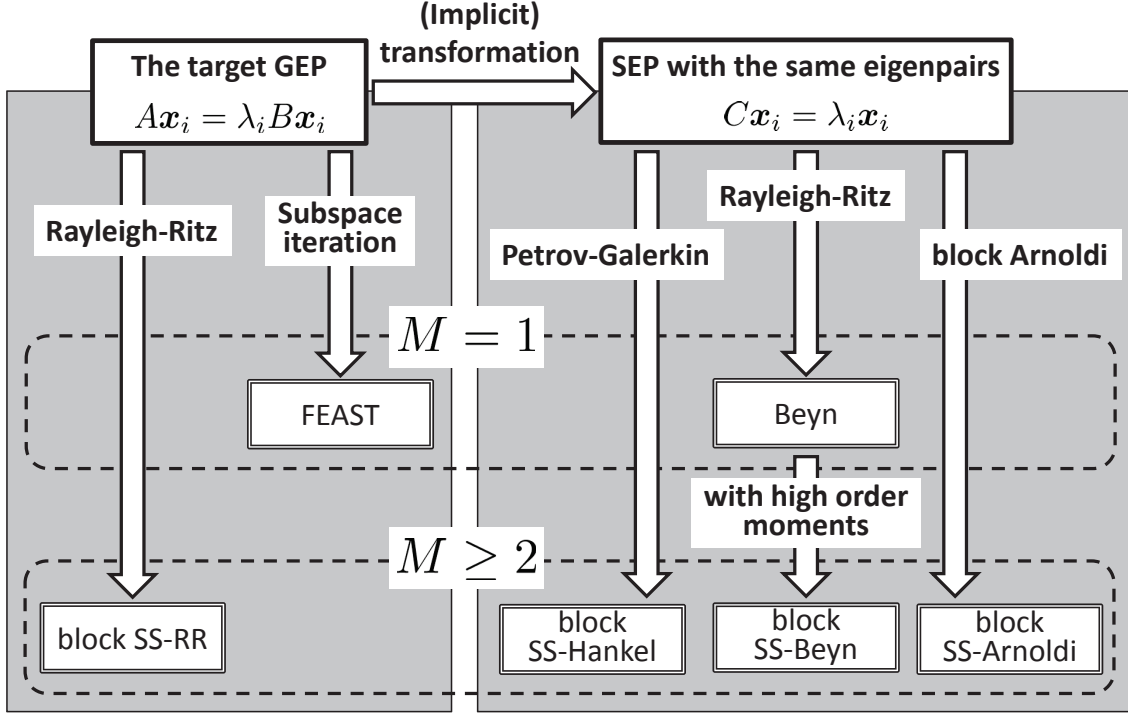


Fig. 1: A map of the contour integral-based eigensolvers.

4.2 Map of the contour integral-based eigensolvers

As shown in Section 4.1.1, the block SS–RR method and the FEAST eigensolver are based on the Rayleigh–Ritz procedure, which solves the generalized eigenvalue problem $Ax_i = \lambda_i Bx_i$. These methods use subspaces $\mathcal{R}(\hat{S})$ and $\mathcal{R}(\hat{S}_0)$, respectively. The FEAST eigensolver constitutes as a simplified algorithm of the block SS–RR method with $M = 1$ and no orthogonalization of the basis. Instead, the FEAST eigensolver presupposes an iteration based on an accelerated subspace iteration. Here, we note that the block SS–RR method can also use an iteration technique for improving accuracy as demonstrated in [11, 17].

In constant, as shown in Section 4.1.2, the block SS–Hankel, block SS–Arnoldi and Beyn methods can be regarded as projection methods for solving the standard eigenvalue problem (21) instead of $Ax_i = \lambda_i Bx_i$. The block SS–Hankel method is a Petrov–Galerkin-type method with $\mathcal{R}(\hat{S})$, the block SS–Arnoldi method is a block Arnoldi method with $\mathcal{R}(\hat{S}) = \mathcal{K}_M^\square(C, \hat{S}_0)$ and the Beyn method is a Rayleigh–Ritz-type method with $\mathcal{R}(\hat{S}_0)$. Note that because these methods are based on Theorems 8 and 9, (z_j, ω_j) should satisfy condition (10).

Since the block SS–Hankel, block SS–RR method and block SS–Arnoldi methods use $\mathcal{R}(\hat{S})$ as the subspace, the maximum dimension of the subspace is LM . In constant, the FEAST eigensolver and the Beyn method use the subspace $\mathcal{R}(\hat{S}_0)$, whose maximum dimension is L ; that is, $\mathcal{R}(\hat{S}_0)$ can not be larger than the number L of right-hand sides of linear systems in each quadrature point. Therefore, for the same subspace dimension, the FEAST eigensolver and the Beyn method should incur larger computational costs than the block SS–Hankel, block SS–RR and block SS–Arnoldi methods for solving the linear systems with multiple right-hand sides.

A map of the contour integral-based eigensolvers is presented in Fig. 1.

Algorithm 7 A block SS–Beyn method

Input: $L, M, N \in \mathbb{N}, V \in \mathbb{C}^{n \times L}, (z_j, \omega_j)$ for $j = 1, 2, \dots, N$

Output: Approximate eigenpairs $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i)$ for $i = 1, 2, \dots, \hat{m}$

- 1: Compute $\hat{S}_k = \sum_{j=1}^N \omega_j z_j^k (z_j B - A)^{-1} B V$,
and set $\hat{S} = [\hat{S}_0, \hat{S}_1, \dots, \hat{S}_{M-1}]$, $\hat{S}_+ = [\hat{S}_1, \hat{S}_2, \dots, \hat{S}_M]$
 - 2: Compute SVD of \hat{S} : $\hat{S} = [U_1, U_2][\Sigma_1, O; O, \Sigma_2][W_1, W_2]^H$
 - 3: Compute eigenpairs (θ_i, \mathbf{t}_i) of $U_1^H S_+ W_1 \Sigma_1^{-1} \mathbf{t}_i = \theta_i \mathbf{t}_i$,
and compute $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i) = (\theta_i, U_1 \mathbf{t}_i)$ for $i = 1, 2, \dots, \hat{m}$
-

4.3 Proposal for a block SS–Beyn method

As mentioned above, one iteration of the FEAST eigensolver is a simplified version of the block SS–RR method with $M = 1$ and no orthogonalization. In constant, a derivative of the Beyn method with $M \geq 2$ has not been proposed. Although, this paper mainly aim to analyze the relationships among these methods and provide a map, we also propose an extension of the Beyn method to $M \geq 2$ as with the block SS–Hankel, block SS–RR and block SS–Arnoldi methods.

As shown in Sections 2.2.5 and 4.1.2, from the relation $\hat{S}_1 = C\hat{S}_0$ and a singular value decomposition of \hat{S}_0 , we can derive a small size eigenvalue problem (9) of the Beyn method. To extend the Beyn method with $\mathcal{R}(\hat{S})$, we introduce

$$\hat{S}_+ := [\hat{S}_1, \hat{S}_2, \dots, \hat{S}_M] = C\hat{S}.$$

Then, using a singular value decomposition of \hat{S}

$$\hat{S} = [U_1, U_2] \begin{bmatrix} \Sigma_1 & O \\ O & \Sigma_2 \end{bmatrix} \begin{bmatrix} W_1^H \\ W_2^H \end{bmatrix} \approx U_1 \Sigma_1 W_1^H,$$

the reduced eigenvalue problem becomes

$$U_1^H S_+ W_1 \Sigma_1^{-1} \mathbf{t}_i = \theta_i \mathbf{t}_i.$$

In this paper, we call this method as the block SS–Beyn method and is shown in Algorithm 7. Since $U_1^H S_+ W_1 \Sigma_1^{-1} = U_1^H C U_1$, the block SS–Beyn method can be regarded as a Rayleigh–Ritz-type method with $\mathcal{R}(\hat{S})$ rather than $\mathcal{R}(\hat{S}_0)$; see Fig. 1.

Both the block SS–RR method and the block SS–Beyn method are Rayleigh–Ritz-type projection methods with $\mathcal{R}(\hat{S})$. However, since the methods are targeted at different eigenvalue problems, they have different definitions of the residual vector. Therefore, these methods mathematically differ when $B \neq I$. In constant, the block SS–Arnoldi method and the block SS–Beyn method without a low-rank approximation, i.e., $\hat{m} = LM$, are mathematically equivalent.

5 Error analyses of the contour integral-based eigensolvers with an iteration technique

As shown in Section 2.2.3, the FEAST eigensolver is based on the iteration. Other iterative contour integral-based eigensolvers have been designed to improve the accuracy [11, 17].

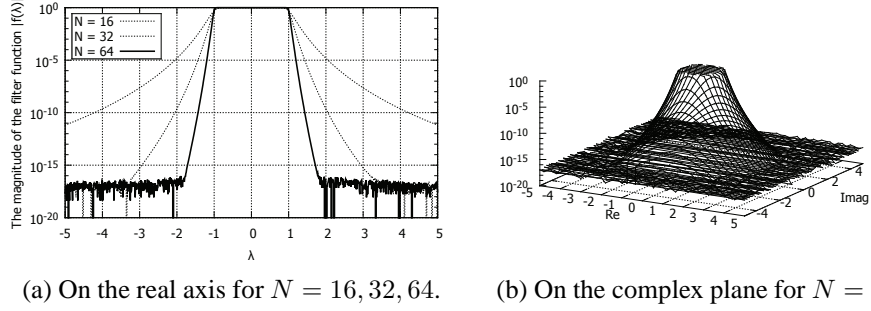


Fig. 2: Magnitude of filter function $|f(\lambda)|$ of the N -point trapezoidal rule for the unit circle region Ω .

The basic concept is the iterative computation of the matrix $\widehat{S}_0^{(\ell-1)}$, from the initial matrix $\widehat{S}_0^{(0)} = V$ as follows:

$$\widehat{S}_0^{(\nu)} := \sum_{j=1}^N \omega_j (z_j B - A)^{-1} B \widehat{S}_0^{(\nu-1)}, \quad \nu = 1, 2, \dots, \ell - 1. \quad (23)$$

The matrices $\widehat{S}_k^{(\ell)}$ and $\widehat{S}^{(\ell)}$ are then constructed from $\widehat{S}_0^{(\ell-1)}$ as

$$\widehat{S}^{(\ell)} := [\widehat{S}_0^{(\ell)}, \widehat{S}_1^{(\ell)}, \dots, \widehat{S}_{M-1}^{(\ell)}], \quad \widehat{S}_k^{(\ell)} := \sum_{j=1}^N \omega_j z_j^k (z_j B - A)^{-1} B \widehat{S}_0^{(\ell-1)}, \quad (24)$$

and $\mathcal{R}(\widehat{S}_0^{(\ell)})$ and $\mathcal{R}(\widehat{S}^{(\ell)})$ are used as subspaces rather than $\mathcal{R}(\widehat{S}_0)$ and $\mathcal{R}(\widehat{S})$. The ℓ iterations of the FEAST eigensolver can be regarded as a Rayleigh–Ritz-type projection method on $\mathcal{R}(\widehat{S}_0^{(\ell)})$.

From the discussion in Section 3, the matrix $\widehat{S}_0^{(\ell)}$ can be expressed as

$$\widehat{S}_0^{(\ell)} = \left(Q_{1:r} F_{1:r} \widetilde{Q}_{1:r}^H \right)^\ell V.$$

Here, the eigenvalues of the linear operator $\widehat{P} := Q_{1:r} F_{1:r} \widetilde{Q}_{1:r}^H$ are given by

$$f(\lambda_i) := \sum_{j=1}^N \frac{\omega_j}{z_j - \lambda_i}.$$

The function $f(\lambda)$, called the filter function, is used in the analyses of some eigensolvers with diagonalizable matrix pencil [5, 11, 18, 19]. The function $f(\lambda)$ is characterized by $|f(\lambda)| \approx 1$ in the inner region and $|f(\lambda)| \approx 0$ in the outer region. Fig. 2 plots the filter function when Ω is the unit circle and integration is performed by the N -point trapezoidal rule.

Error analyses of the block SS–RR method with the iteration technique (23) and (24) and the FEAST eigensolver in the diagonalizable case were given in [5, 11, 19]. In these error analyses, the block SS–RR method and the FEAST eigensolver were treated as projection methods with the subspaces $\mathcal{R}(\widehat{S})$ and $\mathcal{R}(\widehat{S}_0)$, respectively. In Section 4, we explained that the other contour integral-based eigensolvers are also projection methods with the subspaces

$\mathcal{R}(\hat{S})$ and $\mathcal{R}(\hat{S}_0)$, but were designed to solve the standard eigenvalue problem (21). In this section, we establish the error bounds of the contour integral-based eigensolvers with the iteration technique (23) and (24), omitting the low-rank approximation, in non-diagonalizable cases.

5.1 Error bounds of the block SS–RR method and the FEAST eigensolver in the diagonalizable case

Let $(\lambda_i, \mathbf{x}_i)$ be exact finite eigenpairs of the generalized eigenvalue problem $A\mathbf{x}_i = \lambda_i B\mathbf{x}_i$. Assume that $f(\lambda_i)$ are ordered by decreasing magnitude $|f(\lambda_i)| \geq |f(\lambda_{i+1})|$. Define $\mathcal{P}^{(\ell)}$ and \mathcal{P}_{LM} as orthogonal projectors onto the subspaces $\mathcal{R}(\hat{S}^{(\ell)})$ and the spectral projector with an invariant subspace $\text{span}\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{LM}\}$, respectively. Assume that the matrix $\mathcal{P}_{LM}[V, CV, \dots, C^{M-1}V]$ is full rank. Then, for each eigenvector $\mathbf{x}_i, i = 1, 2, \dots, LM$, there exists a unique vector $\mathbf{s}_i \in \mathcal{K}_M^\square(C, V)$ such that $\mathcal{P}_{LM}\mathbf{s}_i = \mathbf{x}_i$.

In the diagonalizable case, for the error analysis of the block SS–RR method and the FEAST eigensolver, the following inequality was given in [11] and [5, 19] for $M = 1$:

$$\|(I - \mathcal{P}^{(\ell)})\mathbf{x}_i\|_2 \leq \alpha\beta_i \left| \frac{f(\lambda_{LM+1})}{f(\lambda_i)} \right|^\ell, \quad i = 1, 2, \dots, LM, \quad (25)$$

where $\alpha = \|X_r\|_2 \|\tilde{X}_r\|_2$ and $\beta_i = \|\mathbf{x}_i - \mathbf{s}_i\|_2$. Note that, in the diagonalizable case, the linear operator \hat{P} can be expressed as $\hat{P} = X_r f(\Lambda_r) \tilde{X}_r^H$, where $f(\Lambda_r) := \text{diag}(f(\lambda_1), f(\lambda_2), \dots, f(\lambda_r))$. An additional error bound is given in [11]:

$$\|(A_{\mathcal{P}^{(\ell)}} - \lambda_i B_{\mathcal{P}^{(\ell)}})\mathbf{x}_i\|_2 \leq \gamma_i \|(I - \mathcal{P}^{(\ell)})\mathbf{x}_i\|_2 \leq \alpha\beta_i\gamma_i \left| \frac{f(\lambda_{LM+1})}{f(\lambda_i)} \right|^\ell, \quad (26)$$

for $i = 1, 2, \dots, LM$, where $A_{\mathcal{P}^{(\ell)}} := \mathcal{P}^{(\ell)} A \mathcal{P}^{(\ell)}$, $B_{\mathcal{P}^{(\ell)}} := \mathcal{P}^{(\ell)} B \mathcal{P}^{(\ell)}$ and $\gamma_i = \|\mathcal{P}^{(\ell)}(A - \lambda_i B)(I - \mathcal{P}^{(\ell)})\|_2$.

Inequality (25) determines the accuracy of the subspace $\mathcal{R}(\hat{S})$, whereas inequality (26) defines the error bound of the block SS–RR method and the FEAST eigensolver.

5.2 Error bounds of the contour integral-based eigensolvers in the non-diagonalizable case

The constant α in (25) derives from the following inequality for a diagonalizable matrix $G_{\text{diag}} = XDX^{-1}$

$$\|G_{\text{diag}}^\ell\|_2 \leq \|X\|_2 \|D^\ell\| \|X^{-1}\|_2 \leq \|X\|_2 \|X^{-1}\|_2 (\rho(G_{\text{diag}}))^\ell,$$

where $\rho(G_{\text{diag}})$ is the spectral radius of G_{diag} . This inequality is extended to a non-diagonalizable matrix $D_{\text{non}} = XJX^{-1}$ as follows:

$$\|G_{\text{non}}^\ell\|_2 \leq \|X\|_2 \|J^\ell\| \|X^{-1}\|_2 \leq 2\|X\|_2 \|X^{-1}\|_2 \ell^{\eta-1} (\rho(G_{\text{non}}))^\ell,$$

where $\rho(G_{\text{non}})$ is the spectral radius of G_{non} and η is the maximum size of the Jordan blocks. Using this inequality, the error analysis of the contour integral-based eigensolvers in the non-diagonalizable case is governed by

$$\|(I - \mathcal{P}^{(\ell)})\mathbf{x}_i\|_2 \leq \alpha'\beta_i \ell^{\eta-1} \left| \frac{f(\lambda_{LM+1})}{f(\lambda_i)} \right|^\ell, \quad i = 1, 2, \dots, LM, \quad (27)$$

where $\alpha' = 2\|Q_{1:r}\|_2\|\tilde{Q}_{1:r}\|_2$. From (27), the error bound of the block SS–RR method and the FEAST eigensolver in the non-diagonalizable case is given by

$$\|(A_{\mathcal{P}^{(\ell)}} - \lambda_i B_{\mathcal{P}^{(\ell)}})\mathbf{x}_i\|_2 \leq \gamma_i \|(I - \mathcal{P}^{(\ell)})\mathbf{x}_i\|_2 \leq \alpha' \beta_i \gamma_i \ell^{\eta-1} \left| \frac{f(\lambda_{LM+1})}{f(\lambda_i)} \right|^\ell, \quad (28)$$

for $i = 1, 2, \dots, LM$.

The inequality (28) derives from the error bound of the Rayleigh–Ritz procedure for generalized eigenvalue problems $A\mathbf{x}_i = \lambda_i B\mathbf{x}_i$. From the error bound of the Rayleigh–Ritz procedure for standard eigenvalue problems [14, Theorem 4.3], we derive the error bound of the block SS–Arnoldi and block SS–Beyn methods as

$$\|(C_{\mathcal{P}^{(\ell)}} - \lambda_i I)\mathcal{P}^{(\ell)}\mathbf{x}_i\|_2 \leq \gamma' \|(I - \mathcal{P}^{(\ell)})\mathbf{x}_i\|_2 \leq \alpha' \beta_i \gamma' \ell^{\eta-1} \left| \frac{f(\lambda_{LM+1})}{f(\lambda_i)} \right|^\ell, \quad (29)$$

for $i = 1, 2, \dots, LM$, where $C_{\mathcal{P}^{(\ell)}} := \mathcal{P}^{(\ell)} C \mathcal{P}^{(\ell)}$ and $\gamma' = \|\mathcal{P}^{(\ell)} C (I - \mathcal{P}^{(\ell)})\|_2$.

In Addition, let \mathcal{Q} be the oblique projector onto $\mathcal{R}(\tilde{S}^{(\ell)})$ and orthogonal to $\mathcal{R}(\tilde{S})$. Then, from the error bound of the Petrov–Galerkin-type projection method for standard eigenvalue problems [14, Theorem 4.7], the error bound of the block SS–Hankel method is derived as follows:

$$\|(C_{\mathcal{P}^{(\ell)}}^{\mathcal{Q}} - \lambda_i I)\mathcal{P}^{(\ell)}\mathbf{x}_i\|_2 \leq \gamma''_i \|(I - \mathcal{P}^{(\ell)})\mathbf{x}_i\|_2 \leq \alpha' \beta_i \gamma''_i \ell^{\eta-1} \left| \frac{f(\lambda_{LM+1})}{f(\lambda_i)} \right|^\ell, \quad (30)$$

for $i = 1, 2, \dots, LM$, where $C_{\mathcal{P}^{(\ell)}}^{\mathcal{Q}} := \mathcal{Q} C \mathcal{P}^{(\ell)}$ and $\gamma''_i = \|\mathcal{Q}(C - \lambda_i I)(I - \mathcal{P}^{(\ell)})\|_2$.

Error bounds (28), (29) and (30) indicate that given a sufficiently large subspace, i.e., $|f(\lambda_{LM+1})/f(\lambda_i)|^\ell \approx 0$, the contour integral-based eigensolvers can obtain the accurate target eigenpairs even if some eigenvalues exist outside but near the region and the target matrix pencil is non-diagonalizable.

6 Numerical experiments

This paper mainly aims to analyze the relationships among the contour integral-based eigensolvers and to map these relationships; although, in this section, the efficiency of the block SS–Hankel, block SS–RR, block SS–Arnoldi and block SS–Beyn methods are compared in numerical experiments with $M = 1, 2, 4, 8$ and 16 .

These methods compute 1000 eigenvalues in the interval $[-1, 1]$ and the corresponding eigenvectors of a real symmetric generalized eigenvalue problem with 20000 dimensional dense and random matrices. Γ is an ellipse with center 0 and major and minor axes 1 and 0.1, respectively. The parameters are $(L, M) = (4096, 1), (2048, 2), (1024, 4), (512, 8), (256, 16)$ (note that $LM = 4096$) and $N = 32$. Because of a symmetry of the problem, the number of required linear systems is $N/2 = 16$. For the low-rank approximation, we used singular values σ_i satisfying $\sigma_i/\sigma_1 \geq 10^{-14}$ and their corresponding singular vectors, where σ_1 is the largest singular value.

The numerical experiments were carried out in double precision arithmetic on 8 nodes of COMA at CCS, University of Tsukuba. COMA has two Intel Xeon E5-2670v2 (2.5 GHz) and two Intel Xeon Phi 7110P (61 cores) per node. In these numerical experiments, we used

Table 1: Computational results of the block SS–Hankel, block SS–RR, block SS–Arnoldi and block SS–Beyn methods with $M = 1, 2, 4, 8$ and 16 .

Method	M	\hat{m}	Time [sec.]				residual norm	
			t_{LU}	t_{Solve}	t_{Other}	t_{Total}	$\max_{\lambda_i \in \Omega} \ \mathbf{r}_i\ _2$	$\min_{\lambda_i \in \Omega} \ \mathbf{r}_i\ _2$
SS–Hankel	1	1274	126.47	97.80	41.57	265.84	1.72×10^{-14}	3.06×10^{-15}
	2	1291	126.38	49.02	28.74	204.14	1.12×10^{-12}	2.72×10^{-15}
	4	1320	126.46	25.40	25.93	177.78	2.15×10^{-14}	3.16×10^{-15}
	8	1419	126.33	13.53	26.39	166.25	1.31×10^{-11}	1.66×10^{-14}
	16	2206	126.24	7.65	32.41	166.30	1.64×10^{-06}	1.59×10^{-11}
SS–RR	1	1283	126.45	97.27	38.62	262.33	1.34×10^{-13}	1.05×10^{-13}
	2	1292	126.31	48.77	38.84	213.92	1.35×10^{-13}	9.56×10^{-14}
	4	1304	126.34	25.22	38.49	190.05	1.73×10^{-13}	9.89×10^{-14}
	8	1340	126.33	13.46	38.78	178.57	5.53×10^{-13}	1.16×10^{-13}
	16	1461	126.49	7.65	40.84	174.98	1.34×10^{-11}	1.24×10^{-13}
SS–Arnoldi	1	4096	125.96	97.13	94.58	317.66	4.72×10^{-08}	4.46×10^{-12}
	2	4096	126.43	48.84	62.11	237.37	5.24×10^{-08}	1.99×10^{-13}
	4	4096	126.13	25.20	52.61	203.94	2.64×10^{-08}	5.24×10^{-13}
	8	4096	126.23	13.46	49.32	189.02	9.05×10^{-09}	8.80×10^{-13}
	16	4096	126.35	7.63	54.41	188.38	9.31×10^{-07}	7.70×10^{-13}
SS–Beyn	1	1283	126.17	97.24	32.63	256.05	1.34×10^{-13}	1.06×10^{-13}
	2	1292	126.48	48.76	32.14	207.37	1.36×10^{-13}	9.58×10^{-14}
	4	1304	126.22	25.22	31.25	182.69	1.74×10^{-13}	9.91×10^{-14}
	8	1340	126.21	13.44	31.09	170.74	5.54×10^{-13}	1.16×10^{-13}
	16	1461	126.45	7.65	32.25	166.35	1.90×10^{-10}	1.25×10^{-13}

only the CPU part. The algorithms were implemented in Fortran 90 and MPI, and executed with $8 [\text{node}] \times 2 [\text{process/node}] \times 8 [\text{thread/process}]$.

The numerical results are presented in Table 1. First, we consider the computation time. The computation times of the LU factorization, forward and back substitutions and the other computation time including the singular value decomposition and orthogonalization are denoted by t_{LU} , t_{Solve} , t_{Other} , respectively. The total computation time is also denoted by t_{Total} . We observe, from Table 1, that the most time-consuming part is to solve linear systems with multiple-right hand sides ($t_{\text{LU}} + t_{\text{Solve}}$). In particular, t_{Solve} is much larger for $M = 1$ than for $M = 16$, because the number of right-hand sides for $M = 1$ is 16 times larger than for $M = 16$. Consequently, t_{Total} increases with decreasing M .

We now focus on t_{Other} . The block SS–Arnoldi method consumes much greater t_{Other} than the other methods because its current version applies no low-rank approximation technique to reduce the computational costs and improve the stability [9]. Comparing the $M = 1$ and $M = 16$ computations by the block SS–Hankel, block SS–RR and block SS–Beyn methods, we observe that the numerical rank \hat{m} and t_{Other} are both smaller for $M = 1$ than for $M = 16$. In addition, the block SS–Hankel method consumes smallest t_{Other} among tested methods, because it operates a no matrix orthogonalization.

Finally, we consider the accuracy of the computed eigenpairs. The block SS–Hankel and block SS–Arnoldi methods are less accurate than the other methods, specifically for $M = 16$. This result is attributed to no matrix orthogonalization in the block SS–Hankel method, and

to no low-rank approximation in the block SS–Arnoldi method. On the other hand, the block SS–RR and block SS–Beyn methods show high accuracy even for $M = 16$.

7 Conclusions

In this paper, we analyzed and mapped the mathematical relationships among the algorithms of the typical contour integral-based eigensolvers for solving generalized eigenvalue problems (1): the block SS–Hankel method, the block SS–RR method, the FEAST eigensolver, the block SS–Arnoldi method and the Beyn method. We found that the block SS–RR method and the FEAST eigensolver are projection methods for $Ax_i = \lambda_i Bx_i$, whereas the block SS–Hankel, block SS–Arnoldi and Beyn methods are projection methods for the standard eigenvalue problem $Cx_i = \lambda_i x_i$. From the map of the algorithms, we also extended the existing Beyn method to $M \geq 2$. Our numerical experiments indicated that increasing M reduces the computational costs (relative to $M = 1$).

In future, we will compare the efficiencies of these methods in solving large, real-life problems. We also plan to analyze the relationships among contour integral-based nonlinear eigensolvers.

Acknowledgements

The authors would like to thank Dr. Kensuke Aishima, The University of Tokyo for his valuable comments. This research was supported partly by Interdisciplinary Computational Science Program in CCS, University of Tsukuba, Strategic Programs for Innovative Research (SPIRE) Field 5 “The origin of matter and the universe”, JST/CREST and KAKENHI (Grant Nos. 25286097, 25870099).

References

- [1] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, K. Kimura, A numerical method for nonlinear eigenvalue problems using contour integrals, *JSIAM Letters*, **1**(2009), 52–55.
- [2] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, K. Kimura, A numerical method for polynomial eigenvalue problems using contour integral, *Japan J. Indust. Appl. Math.*, **27**(2010), 73–90.
- [3] W.-J. Beyn, An integral method for solving nonlinear eigenvalue problems, *Lin. Alg. Appl.*, **436**(2012), 3839–3863.
- [4] M. H. Gutknecht, Block Krylov space methods for linear systems with multiple right-hand sides: an introduction, in: *proc. of Modern Mathematical Models, Methods and Algorithms for Real World Systems* (A.H. Siddiqi, I.S. Duff, and O. Christensen, eds.), 420–447, Anamaya Publishers, New Delhi, India, 2007.
- [5] S. Güttel, E. Polizzi, T. Tang, G. Viaud, Zolotarev quadrature rules and load balancing for the FEAST eigensolver, *arXiv:1407.8078*.

- [6] T. Ikegami, T. Sakurai, U. Nagashima, A filter diagonalization for generalized eigenvalue problems based on the Sakurai-Sugiura projection method, Technical Report of Department of Computer Science, University of Tsukuba (CS-TR), (2008), CS-TR-08-13.
- [7] T. Ikegami, T. Sakurai, U. Nagashima, A filter diagonalization for generalized eigenvalue problems based on the Sakurai-Sugiura projection method, *J. Comput. Appl. Math.*, **233**(2010), 1927–1936.
- [8] T. Ikegami, T. Sakurai, Contour integral eigensolver for non-Hermitian systems: a Rayleigh-Ritz-type approach, *Taiwan. J. Math.*, **14**(2010), 825–837.
- [9] A. Imakura, L. Du, T. Sakurai, A block Arnoldi-type contour integral spectral projection method for solving generalized eigenvalue problems, *Applied Mathematics Letters*, **32**(2014), 22–27.
- [10] A. Imakura, L. Du, T. Sakurai, Communication-Avoiding Arnoldi-type contour integral-based eigensolver (in Japanese), in: *proc. of Annual meeting of JSIAM*, 2014.
- [11] A. Imakura, L. Du, T. Sakurai, Error bounds of Rayleigh–Ritz type contour integral-based eigensolver for solving generalized eigenvalue problems, *Numer. Alg.*, (accepted).
- [12] P. Kravanja, T. Sakurai, M. van Barel, On locating clusters of zeros of analytic functions, *BIT*, **39**(1999), 646–682.
- [13] E. Polizzi, A density matrix-based algorithm for solving eigenvalue problems, *Phys. Rev. B*, **79**(2009), 115112.
- [14] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, 2nd edition, SIAM, Philadelphia, PA, (2011).
- [15] T. Sakurai, H. Sugiura, A projection method for generalized eigenvalue problems using numerical integration, *J. Comput. Appl. Math.*, **159**(2003), 119–128.
- [16] T. Sakurai, H. Tadano, CIRR: a Rayleigh-Ritz type method with counter integral for generalized eigenvalue problems, *Hokkaido Math. J.*, **36**(2007), 745–757.
- [17] T. Sakurai, Y. Futamura, H. Tadano, Efficient parameter estimation and implementation of a contour integral-based eigensolver, *J. Algorithms Comput. Technol.*, **7**(2014), 249–269.
- [18] G. Schofield, J. R. Chelikowsky, Y. Saad, A spectrum slicing method for the Kohn-Sham problem, *Comput. Phys. Commun.*, **183**(2012), 497–505.
- [19] P. T. P. Tang, E. Polizzi, FEAST as a subspace iteration eigensolver accelerated by approximate spectral projection, *SIAM J. Matrix Anal. Appl.*, **35**(2014), 354–390.
- [20] G. Yin, R. H. Chan, M.-C. Yeung, A FEAST algorithm for generalized non-Hermitian eigenvalue problems, *arXiv:1404.1768*.
- [21] S. Yokota, T. Sakurai, A projection method for nonlinear eigenvalue problems using contour integrals, *JSIAM Letters*, **5**(2013), 41–44.